

CLAIMS:

We claim:

1 1. A method for securing cached data in an enterprise environment, said method
2 comprising the steps of:

3 processing a request to locate data in a query cache;

4 if said data can be located in said query cache, retrieving said data from said
5 query cache, decrypting at least one encrypted portion of said retrieved data, and
6 forwarding said decrypted portion and any remaining unencrypted portion of said
7 retrieved data to a requesting client; and,

8 if said data cannot be located in said query cache, retrieving said data from a
9 back-end data source over a computer communications network, forwarding said
10 retrieved data to said requesting client, encrypting at least a portion of said retrieved
11 data and storing in said query cache said encrypted portion and any remaining
12 unencrypted portion.

1 2. The method of claim 1, wherein said encrypting and decrypting steps utilize an
2 encryption key stored in a hardware security module (HSM).

3 3. The method of claim 1, wherein said processing step comprises:
4 locating within said request a key;
5 subjecting said key to a one-way hashing function; and,

6 comparing said hashed key to individual one-way hashed keys in said query
7 cache, said comparison determining whether said data can be located in said query
8 cache.

1 4. A method for securing cached data in an edge-deployed query cache, said
2 method comprising the steps of:

3 configuring an encryption engine with an encryption system selected from the
4 group consisting of a hardware security module (HSM), a software encryption module,
5 or a combined hardware and software encryption system;

6 associating the edge-deployed query cache with said encryption engine; and,

7 configuring the edge-deployed query cache both to encrypt entries for storage in
8 the edge-deployed query cache using said configured encryption engine the edge-
9 deployed query cache retrieving said entries for storage from a back-end data source
10 prior to said encryption, and also to decrypt entries retrieved from the edge-deployed
11 query cache using said configured encryption engine.

1 5. A secured query cache system comprising:

2 a query cache disposed in an edge server; and,

3 an encryption engine communicatively linked to said edge server, said encryption
4 engine having a configuration for encrypting entries to said query cache and decrypting
5 entries retrieved from said query cache.

1 6. The secured query cache system of claim 4, wherein said encryption engine
2 comprises:

3 a modular interface for accepting interchangeable encryption configurations, said
4 configurations comprising one of a hardware security module, and a software
5 encryption component.

6 7. The system of claim 6, wherein said encryption module implements the
7 JAVA(TM) Cryptography Extension.

1 8. A machine readable storage having stored thereon a computer program for
2 securing cached data in an enterprise environment, said computer program comprising
3 a routine set of instructions for causing the machine to perform the steps of:

4 processing a request to locate data in a query cache;
5 if said data can be located in said query cache, retrieving said data from said
6 query cache, decrypting at least one encrypted portion of said retrieved data, and
7 forwarding said decrypted portion and any remaining unencrypted portion of said
8 retrieved data to a requesting client; and,

9 if said data cannot be located in said query cache, retrieving said data from a
10 back-end data source over a computer communications network, forwarding said
11 retrieved data to said requesting client, encrypting at least a portion of said retrieved
12 data and storing in said query cache said encrypted portion and any remaining
13 unencrypted portion.

1 9. The machine readable storage of claim 9, wherein said encrypting and
2 decrypting steps utilize an encryption key stored in a hardware security module (HSM).

1 10. The machine readable storage of claim 8, wherein said processing step
2 comprises:

3 locating within said request a key;

4 subjecting said key to a one-way hashing function; and,
5 comparing said hashed key to individual one-way hashed keys in said query
6 cache, said comparison determining whether said data can be located in said query
7 cache.